Capturing Architectural Knowledge with Architectural Patterns

Harrison, Neil B.; Avgeriou, Paris; Zdun, Uwe

Published in:
EPRINTS-BOOK-TITLE

Focus Group Report: Capturing Architectural Knowledge with Architectural Patterns

Neil B. Harrison[1], Paris Avgeriou[2], and Uwe Zdun[3]

[1] Department of Computer Science and Engineering, Utah Valley State College, Orem, Utah, USA, harrisne@uvsc.edu
[2] Department of Mathematics and Computing Science, University of Groningen, Groningen, the Netherlands, paris@cs.rug.nl
[3] Distributed Systems Group, Information Systems Institute, Vienna University of Technology, Vienna, Austria, zdun@acm.org

Introduction

In the field of Software Architecture, there is a shift away from the paradigm of describing Components and Connectors, and towards regarding Architecture as a set of design decisions. Unfortunately such decisions are rarely documented explicitly in the Software Architecture Document, which leads to inevitable loss of valuable architectural knowledge. This in turn results in increasing the complexity of the system, and hindering its ability to manage continuous change.

The choice and application of architectural patterns during software architecting is in fact a decision-making process, and the patterns themselves represent one of the most significant type of architectural design decisions. These decisions aim at satisfying the architectural key drivers and addressing the stakeholders concerns. In fact, these decisions are very rich in semantics, since architectural patterns by definition are accompanied by the system context, the problem forces, the way to resolve the forces (or not), the rationale, and the benefits and liabilities. Therefore architectural patterns can be considered as first-class candidates for capturing architectural knowledge: they make architectural design decisions explicit, they provide rich context for these decisions and they facilitate sharing these decisions with the relevant stakeholders.

It is clear that an understanding of the decisions made in designing an architecture is crucial to understanding and maintaining the system effectively. But how are these decisions made, and more particularly, how are they documented during the course of architecture design? We believe that teams can use architecture patterns to capture and record architectural design decisions during design. Since decisions are not explicitly documented, patterns can serve as the appropriate medium for mining the most significant decisions.

In order to test this premise, we observed architects in action. In this experiment, we observed two groups as they designed the architecture for a system. We noted the decisions they made, and their use of architecture patterns.

The focus group had the following format:

- Short introduction to the topic by organizers, and discussion about capturing architectural knowledge with patterns.

- Handing out a list of architectural patterns and the description of a system to be designed.
- Formation of two groups where each groups works on the architectural design of the system using patterns and documenting their decisions.
- Wrap-up and collection of results

The goal of this focus group was to examine how architectural patterns can serve as significant architectural design decisions and thus provide reusable, shareable architectural knowledge. We aimed at bringing together practitioners, researchers, developers, etc. to share their experiences and insights with taking architectural design decisions by choosing and applying architectural patterns. The participants demonstrated in practice the architectural design of a non-trivial system according to architectural patterns and accordingly made their decisions explicit.

The Experiment

We divided the participants into two groups. One group consisted of highly experienced software developers and architects averaging over ten years development experience and over four years architecture experience. The other group had little experience in professional software development and architecture; they were nearly all graduate students. Both groups were familiar with architecture patterns, though the inexperienced group had little experience using them in practice.

Both groups were assigned to design a web-based application similar to a well-known real-world application. Even though the teams had less than three hours in which to work, the application was large and complex. It was our intent that they wrestle with design decisions; not that they finish. We also wanted to observe which design issues tended to come up earliest.

The requirements were purposely somewhat general. They included both functional and non-functional requirements. We considered providing a list of architecture patterns, but decided that such a list might tend to skew thinking toward usage of the patterns.

The participants were asked to record the decisions they made as they made them. Near the end of the exercise, they were asked to review their work and identify the decisions they had made. Then they were asked to present their work to each other, in terms of the decisions they had made. Note that they were asked three separate times to identify the decisions they made.

Results

We observed the groups as they designed the system. Each group began by discussing requirements and selecting a few of them to begin addressing. They explored alternative design approaches to the selected requirements, and then discussed other requirements and associated design approaches. They continued in this manner for about two hours. They used white boards to write or draw ideas and organize information.

We then asked them to review their work, and to summarize and record the decisions they had made. We further asked the groups to present the decisions they made to each other. However, neither group explicitly identified any decisions, either during their own wrap-up or when presenting to the other group. Instead, they presented their architectural structures they had so far created. They did present the architectural patterns they used – and knew that they had done so.

Decisions Made

We noted that although the time was limited, both groups made several architectural decisions during the course of the workshop. The decisions were, as expected, mainly decisions about approaches to the requirements given for the project. These included decisions about functional requirements, such as adopting a thin client model, and about non-functional requirements, such as approaches to replication for enhanced reliability and availability of the application.

Perhaps the earliest significant decision each group made was which requirements to begin with. This decision implied that certain requirements were important to address early. Neither group identified this as a decision, and it appeared they did not even recognize that it was a decision.

In several instances, the groups made decisions to explicitly defer design and/or decisions until later. Such decisions would not be enduring decisions that would be used later to discover the architecture, but would instead serve as reminders to designers and developers not to forget certain issues.

Both groups made decisions about the overall structure of the system quite quickly. For the experienced group, these decisions were based on prior systems with similar requirements. On the other hand, the inexperienced group decided on an overall structure based on lack of experience with any alternatives. On the one hand, the decision was based on extensive experience; on the other, it was based on lack of experience.

Decisions Recorded

It was significant to note that no decisions were explicitly recorded. Neither the highly experienced group nor the inexperienced group recorded decisions explicitly. In spite of the fact that we asked the groups to record decisions, they did not. When we asked them to review their work and identify the decisions made, they simply reviewed the work they had done up to that point. They did not identify any decisions as such. When they presented their work, they discussed what they had done, but did not identify any explicit decisions. This was equally true for both groups, regardless of the experience level.

Why were the groups unwilling or unable to explicitly record or even identify design decisions? It is probably a combination of the following factors:

1. They were in the flow of design, and did not want to interrupt the design flow to record decisions. It was certainly true that the groups were in a

design flow that they would prefer not to break, but this does not account for the fact that they did not record decisions when they reviewed their work.

2. The participants did not consider it important to record their decisions. In this study, we asked them to record their decisions; it appeared that they immediately got caught up in design activities, which appeared more relevant to them, and ignored our requests.
3. They may not have known how to document decisions, or were stymied by the lack of instruction about a format for recording decisions (we did not suggest any format for recording decisions). It is likely that few, if any of the participants had had any experience with documenting decisions, as documenting architectural decisions is rare in practice. This did not appear to be a significant factor in the lack of documentation of decisions, because neither group made any explicit effort to document decisions, or even questioned how it should be done.
4. They may have thought that the things they wrote down were indeed documentation of decisions. They drew a few sparse structural diagrams, wrote a few pattern names, and noted one or two issues to be considered later. They referred to these writings during their presentations to each other, which indicates that they considered that they had captured important information about the architecture. However, it did not fully capture the decisions they had made. This may be related to the next point.
5. The participants may not have recognized that they were making decisions; the decisions were more implicit than overt. This appeared to be a significant factor; probably the most significant one. The lack of decision identification during the review and presentations supports this premise. In addition, as will be discussed below, use of architecture patterns was much more implicit than explicit. Designing software architecture is a highly creative, intuitive, and emergent process, in which the process is not always apparent.
6. The lack of time was a factor, at least for noting which patterns were used. The experienced group noted that they had not really begun to identify patterns to be used. With more time, they expected to have been able to explore patterns and note which they used.

Architecture Patterns Used

Both groups used architecture patterns. In both cases, the use of patterns was preliminary and light. However, the groups indicated that it was simply too early to use architecture patterns extensively; pattern use would become more pronounced over time. In spite of the preliminary use of patterns, we were able to identify trends in the use of patterns.

Patterns were identified by name, and these names were a part of the conversation, and conveyed significant chunks of information. The use of pattern names was limited in the inexperienced group, because of their limited familiarity with the body of architecture patterns (mainly patterns from POSA and GOF). On the other hand, the

experienced group could name the patterns they used. The way this happened was that they examined their design and found patterns that they had somewhat unconsciously incorporated into the design.

Patterns were not explicitly used in the design of the architecture. Instead, the patterns emerged during the course of design. Neither group explicitly decided to use or not use a particular pattern. Instead, people were working to craft a solution to the requirements. As they did so, the patterns to be used gradually became apparent. For example, the groups partitioned the system into layers, and the Layers pattern emerged. As they partitioned the responsibilities of the system, the Client-Server pattern also emerged. Both groups were making decisions that would tend to lead toward the Broker pattern as time ran out.

The emergent use of patterns is significant in that patterns embody sets of decisions that are made over the course of architectural design. As we have seen, these decisions are generally not explicitly recorded. However, the architectural patterns used can easily be identified by name. In this way the architecture pattern becomes a record of many of the significant design decisions made.

Conclusions

This study is consistent with the authors' experience. However, additional validation, using more extensive projects is desirable. Observation of architecture activities in an industrial setting would be very useful.

It would also be useful to study how architecture patterns can be used by people learning the system to understand which architectural decisions were made, and why they were made.

The question of rationale; why a decision was made, is interesting. Patterns typically have "forces", which provide deeper insight into the problem. Forces often also give information that leads to an understanding of the rationale behind the solution. Patterns also often give information about tradeoffs (e.g., space for time), which is additional rationale information. It would be instructive to learn how people learning an architecture use this additional information in a pattern to understand the architecture.